Neural Homomorphic Vocoder

Zhijun Liu, Kuan Chen, Kai Yu

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering, AI Institute
Shanghai Jiao Tong University, Shanghai
{zhijunliu, azraelkuan, kai.yu}@sjtu.edu.cn

Abstract

In this paper, we propose the neural homomorphic vocoder (NHV), a source-filter model based neural vocoder framework. NHV synthesizes speech by filtering impulse trains and noise with linear time-varying (LTV) filters. A neural network controls the LTV filters by estimating complex cepstrums of time-varying impulse responses given acoustic features. The proposed framework can be trained with a combination of multi-resolution STFT loss and adversarial loss functions. Due to the use of DSP-based synthesis methods, NHV is highly efficient, fully controllable and interpretable. A vocoder was built under the framework to synthesize speech given log-Mel spectrograms and fundamental frequencies. While the model cost only 15 kFLOPs per sample, the synthesis quality remained comparable to baseline neural vocoders in both copy-synthesis and text-to-speech.

Index Terms: speech synthesis, source-filter model, harmonic-plus-noise model, waveform model

1. Introduction

Generative neural networks have obtained tremendous success in generating high-fidelity speech and other audio signals. Audio generation models conditioned on speech features such as log-Mel spectrograms can be used as vocoders. Neural vocoders have greatly improved the synthesis quality of modern text-to-speech systems [1, 2]. Auto-regressive models, including WaveNet [3] and WaveRNN [4], generate audio a sample at a time conditioned on all previously generated samples. Flow-based models, including Parallel WaveNet [5], ClariNet [6], WaveGlow [7] and FloWaveNet [8], generate audio samples in parallel with invertible transformations. GAN based models, including GAN-TTS [9], Parallel WaveGAN [10], and Mel-GAN [11], are also capable of parallel generation. Instead of being trained with maximum likelihood, they are trained with adversarial loss functions.

Neural vocoders can be designed to include speech synthesis models in order to reduce computational complexity and further improve synthesis quality. Many models aim to improve source signal modeling in a source-filter model, including LPC-Net [12], GELP [13], GlotGAN [14]. They only generate source signals (e.g., linear prediction residual signal) with neural networks while offloading spectral shaping to time-varying filters. Instead of improving source signal modeling, the neural source-filter (NSF) [15, 16] framework replaces linear filters in the classical model with convolutional neural network based filters. NSF can synthesize waveform by filtering a simple sine-based excitation signal [15]. Neural audio synthesis with sinusoidal models is also explored recently. DDSP [17] proposes to synthesis audio by controlling a Harmonic plus Noise model with a neural network. In DDSP, the harmonic component is synthesized with additive synthesis where sinusoids with time-varying amplitudes are added. And the noise component is synthesized with linear time-varying filtered noise. DDSP has been proved successful in modeling musical instruments. In this work, we further explore the integration of DSP components in neural vocoders.

We propose a novel neural vocoder framework called neural homomorphic vocoder, which synthesizes speech with source-filter models controlled by a neural network. We demonstrate that with a shallow CNN containing 0.6 million parameters, we can build a neural vocoder capable of reconstructing high-quality speech from log-Mel spectrograms and fundamental frequencies. While the computational complexity is more than 100 times lower compared to baseline systems, the quality of generated speech remains comparable. Audio samples and further information are provided in the online supplement[1]. We highly recommend readers to listen to the audio samples.

2. Neural homomorphic vocoder

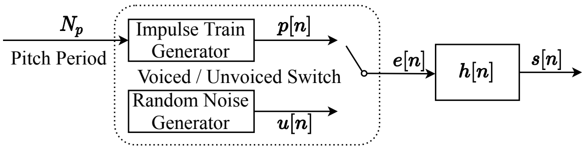

Figure 1: *A simplified source-filter model in discrete time. $e[n]$ is source signal. $s[n]$ is speech.*

The source-filter model is a widely applied linear model of speech production and synthesis [18]. A simplified version of the source-filter model is demonstrated in figure 1. The linear filter $h[n]$ describes the combined effect of glottal pulse, vocal tract, and radiation in speech production. The source signal $e[n]$ is assumed to be either a periodic impulse train $p[n]$ in voiced speech, or noise signal $u[n]$ in unvoiced speech. In practice, $e[n]$ can be a multi-band mixture of impulse and noise [19–21]. N_p is time-varying. And $h[n]$ is replaced with a linear time-varying filter.

In neural homomorphic vocoder (NHV), a neural network controls linear time-varying (LTV) filters in source-filter models. Similar to the Harmonic plus Noise model, NHV generates harmonic and noise components separately. The harmonic component, which contains periodic vibrations in voiced sounds, is modeled with LTV filtered impulse trains. The noise component, which includes background noise, unvoiced sounds, and the stochastic component in voiced sounds, is modeled with LTV filtered noise.

Kai Yu is the corresponding author.

[1] https://zjlww.github.io/is2020/

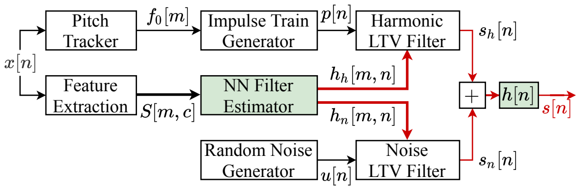

Figure 2: *Illustration of NHV during inference. Gradients are propagated backward along red lines. Green boxes contain trainable parameters.*

In the following discussion, the original speech signal x and reconstructed signal s are divided into non-overlapping frames with frame length L. We define m as the frame index, n as the the discrete time index, and c as the feature index. The total number of frames M and total number of sampling points N follow $N = M \times L$. In f_0, S, h_h, h_n, $0 \leq m < M - 1$. x, s, p, u, s_h, s_n are finite duration signals, in which $0 \leq n < N - 1$. Impulse responses h_h, and h_n are infinitely long, in which $n \in \mathbb{Z}$. Impulse response h is causal, in which $n \in \mathbb{Z}$ and $n \geq 0$.

The speech synthesis process is illustrated in figure 2. First, the impulse train $p[n]$ is generated from frame-wise fundamental frequency $f_0[m]$. And the noise signal $u[n]$ is sampled from a Gaussian distribution. Then, the neural network estimates impulse responses $h_h[m,n]$ and $h_n[m,n]$ in each frame, given the log-Mel spectrogram $S[m,c]$. Next, the impulse train $p[n]$ and the noise signal $u[n]$ are filtered by LTV filters to obtain the harmonic component $s_h[n]$ and the noise component $s_n[n]$. Finally, $s_h[n]$ and $s_n[n]$ are added together and filtered by a trainable causal FIR filter $h[n]$, as proposed in DDSP [17].

In order to train the neural network, multi-resolution STFT loss L_R, and adversarial losses L_G and L_D are computed from $x[n]$ and $s[n]$, as illustrated in figure 3. Since LTV filters are fully differentiable, gradients can propagate back to the NN filter estimator.

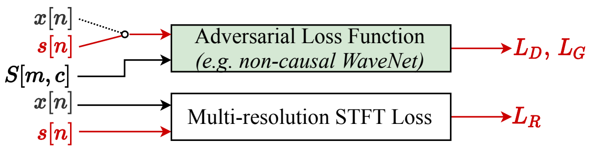
$x[n]$
$s[n]$
$S[m,c]$
Adversarial Loss Function
(e.g. non-causal WaveNet)
L_D, L_G
$x[n]$
$s[n]$
Multi-resolution STFT Loss
L_R

Figure 3: *Illustration of the loss functions used to train NHV.*

In the following sections, we further describe different components in the NHV framework.

2.1. Impulse train generator

Many methods [22, 23] exist for generating alias-free discrete time impulse trains. Additive synthesis is one of the most accurate methods. As described in equation (1), we can use a low-passed sum of sinusoids to generate an impulse train. $f_0(t)$ is reconstructed from $f_0[m]$ with zero-order hold or linear interpolation. $p[n] = p(n/f_s)$. f_s is the sampling rate.

$$p(t) = \begin{cases} \sum_{k=1}^{2kf_0(t)<f_s} \cos(\int_0^t 2\pi k & f_0(\tau)\mathrm{d}\tau), \\ & \text{if } f_0(t) > 0 \\ 0, & \text{if } f_0(t) = 0 \end{cases} \tag{1}$$

Additive synthesis can be computationally expensive as it requires summing up about 200 sine functions at the sampling rate. The computational complexity can be reduced with approximations [22, 23]. For example, we can round the fundamental periods to the nearest multiples of the sampling period. In this case, the discrete impulse train is sparse. It can then be generated sequentially, one pitch mark at a time.

2.2. Neural network filter estimator

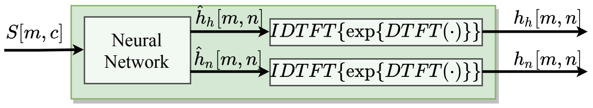

Figure 4: *NN output is defined to be complex cepstrums.*

We propose to use complex cepstrums ($\hat{h}_h$ and $\hat{h}_n$) as the internal description of impulse responses (h_h and h_n). The generation of impulse responses is illustrated in figure 4.

Complex cepstrums describe the magnitude response and the group delay of filters simultaneously. The group delay of filters affects the timbre of speech, as reported in several papers [21, 24, 25] and books [26, 27]. Instead of using linear-phase or minimum-phase filters, NHV uses mixed-phase filters, with phase characteristics learned from the dataset.

Restricting the length of a complex cepstrum is equivalent to restricting the levels of detail in the magnitude and phase response. This gives an easy way to control the filters' complexity. The neural network only predicts low-quefrency coefficients. The high-quefrency cepstrum coefficients are set to zero. In our experiments, two 10 ms long complex cepstrums are predicted in each frame.

In the implementation, the DTFT and IDTFT must be replaced with DFT and IDFT [18]. And IIRs, i.e., $h_h[m,n]$ and $h_n[m,n]$, must be approximated by FIRs. The DFT size should be sufficiently large to avoid serious aliasing. $N = 1024$ is a good choice for our purpose.

2.3. LTV filters and Trainable FIRs

The harmonic LTV filter is defined in equation (3). The noise LTV filter is defined similarly. A trainable causal FIR filter $h[n]$ is applied at the last step in speech synthesis [17]. The convolutions can be carried out in either the time domain or the frequency domain. The filtering process of the harmonic component is illustrated in figure 5.

$$w_L[n] \triangleq \begin{cases} 1, & 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$s_h[n] = \sum_{m=0}^{m<M} (w_L[n-mL] \cdot p[n]) * h_h[m,n] \tag{3}$$

$$s[n] = (s_h[n] + s_n[n]) * h[n] \tag{4}$$

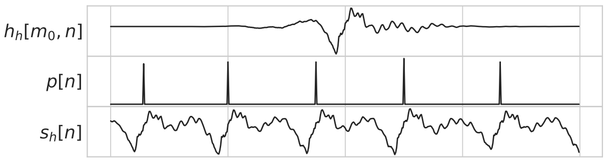

Figure 5: *Signals sampled from a trained NHV model around frame m_0. The figure shows 512 sampling points, or 4 frames. Only one impulse response $h_h[m_0,n]$ from frame m_0 is plotted.*

2.4. Neural network training

2.4.1. Multi-resolution STFT loss

Point-wise loss between $x[n]$ and $s[n]$ can not be applied to train the model, as it requires glottal closure instants (GCIs) in x and s to be fully aligned. Multi-resolution STFT loss is tolerant of phase mismatch in signals [10, 13, 15, 17]. Suppose we have C different STFT configurations, $0 \leq i < C$. Given original signal x, and reconstruction s, their STFT amplitude spectrograms calculated with configuration i are X_i and S_i, each containing K_i values. In NHV, we use a combination of the L^1 norm of amplitude and log-amplitude distances. The reconstruction loss L_R is the sum of all distances under all configurations.

$$L_R = \frac{1}{C} \sum_{i=0}^{C-1} \frac{1}{K_i} \left(\|X_i - S_i\|_1 + \|\log X_i - \log S_i\|_1 \right) \quad (5)$$

We find using more STFT configurations leads to fewer artifacts in output speech. We used Hanning windows with sizes (128, 256, 384, 512, 640, 768, 896, 1024, 1536, 2048, 3072, 4096), with 75% overlap. The FFT sizes are set to twice the window sizes.

2.4.2. Adversarial loss functions

NHV relies on adversarial loss functions with waveform input to learn temporal fine structures in speech signals. Although we do not need adversarial loss functions to guarantee periodicity in NHV, they still help ensure phase similarity between $s[n]$ and $x[n]$. The discriminator should give separate decisions for different short segments in the input signal [9–11]. The discriminator we used in our experiments is a WaveNet conditioned on log-Mel spectrograms. Details of discriminator structure can be found in section 3. We used the hinge loss version of the GAN objective [28, 29] in our experiments.

$$\begin{aligned} L_D =& \mathbb{E}_{x,S} \left[\max\left(0, 1 - D(x, S)\right) \right] + \\ & \mathbb{E}_{f_0,S} \left[\max\left(0, 1 + D(G(f_0, S), S)\right) \right] \end{aligned} \quad (6)$$

$$L_G = \mathbb{E}_{f_0,S} \left[-D(G(f_0, S), S) \right] \quad (7)$$

$D(x, S)$ is the discriminator network. D takes original signal x or reconstructed signal s, and ground truth log-Mel spectrogram S as input. f_0 is the fundamental frequency. S is the log-Mel spectrogram. $G(f_0, S)$ outputs reconstructed signal s. It includes the source signal generation, filter estimation and LTV filtering process in NHV. The discriminator is trained to classify x as real and s as fake by minimizing L_D. And the generator is trained to deceive the discriminator by minimizing L_G.

3. Experiments

To verify the effectiveness of the proposed vocoder framework, we built a neural vocoder and compared its performance in copy synthesis and text-to-speech with various baseline models.

3.1. Corpus and feature extraction

All vocoders and TTS models were trained on the Chinese Standard Mandarin Speech Corpus (CSMSC)[2]. CSMSC contains 10000 recorded sentences read by a female speaker, totaling to 12 hours of high-quality speech, annotated with phoneme sequences, and prosody labels. The original signals were sampled at 48 kHz. In our experiments, audios were downsampled to 22050 Hz. The last 100 sentences were reserved as the test set.

All vocoder models were conditioned on band-limited (40 - 7600 Hz) 80 bands log-Mel spectrograms. The window length used in spectrogram analysis was 512 points (23 ms at 22050 Hz), and the frame shift was 128 points (6 ms at 22050 Hz). We used the REAPER[3] speech processing tool to extract an estimate of the fundamental frequency. The f0 estimations were then refined by StoneMask.

3.2. Model configurations

3.2.1. Details of vocoders

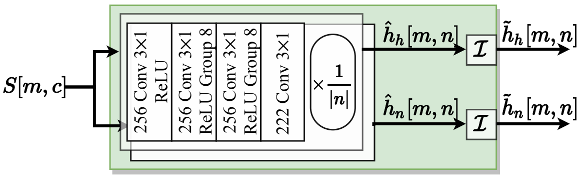

Figure 6: *Network used in experiment. $\mathcal{I}$ is DFT based complex cepstrum inversion. $\tilde{h}_h$ and $\tilde{h}_n$ are DFT approximations of h_h and h_n.*

In the NHV model, two separate 1D convolutional neural networks with the same structure were used for complex cepstrum estimation, as illustrated in figure 6. Note that the outputs of the neural network need to be scaled by $1/|n|$, as natural complex cepstrums decay at least as fast as $1/|n|$,

The discriminator was a non-causal WaveNet conditioned on log-Mel spectrograms with 64 skip and residual channels. The WaveNet contained 14 dilated convolutions. The dilation is doubled for every layer up to 64 and then repeated. The kernel sizes in all layers were 3.

A 50ms exponentially decayed trainable FIR filter was applied to the filtered and mixed harmonic and noise component. We found that this module made the vocoder more expressive and slightly improved perceived quality.

Several baseline systems were used to evaluate the performance of NHV, including an MoL WaveNet [5], two variants of the NSF model, and a Parallel WaveGAN. In order to examine the effect of the adversarial loss, we also trained an NHV model with only multi-resolution STFT loss (NHV-noadv).

The MoL WaveNet [5] pre-trained on CSMSC from ESPNet [30] (csmsc.wavenet.mol.v1) was borrowed for evaluation. The generated audios were downsampled from 24000 Hz to 22050 Hz.

A hn-sinc-NSF [16] model was trained with the released code. We also reproduced the b-NSF model and augmented it with adversarial training (b-NSF-adv). The discriminator in b-NSF-adv contained 10 1D convolutions with 64 channels. All convolutions had kernel size 3, with strides following the sequence (2, 2, 4, 2, 2, 2, 1, 1, 1, 1) in each layer. All layers except for the last one were followed by a leaky ReLU activation with a negative slope set to 0.2. We used STFT window sizes (16, 32, 64, 128, 256, 512, 1024, 2048), and mean amplitude distance instead of mean log-amplitude distance described in the paper [15].

We reproduced the Parallel WaveGAN [10] model. There were several modifications compared to the descriptions in the original paper. The generator was conditioned on log f0, voicing decisions, and log-Mel spectrograms. The same STFT loss configurations in b-NSF-adv were used to train Parallel WaveGAN.

The online supplement contains further details about vocoder training.

[2] https://www.data-baker.com/open_source_en.html

[3] https://github.com/google/REAPER

3.2.2. *Details of the text-to-speech model*

A Tacotron2 [2] was trained to predict log f0, voicing decision, and log-Mel spectrogram from texts. The prosody and phonetic labels in CSMSC were both used to produce text input to Tacotron. NHV, Parallel WaveGAN, b-NSF-adv, and hn-sinc-NSF were used in TTS quality evaluation. We did not fine-tune the vocoders with generated acoustic features.

3.3. Results and analysis

3.3.1. *Performance in copy synthesis*

A MUSHRA test was conducted to evaluate the performance of proposed and baseline neural vocoders in copy synthesis. 24 Chinese listeners participated in the experiment. 18 items unseen during training were randomly selected and divided into three parts. Each listener rated one part out of three. Two standard anchors were used in the test. Anchor35 and Anchor70 represent low-pass filtered original signal with cut-off frequencies of 3.5 kHz and 7 kHz. The box plot of all scores collected is shown in figure 7. The mean MUSHRA scores and their 95% confidence intervals can be found in table 1.

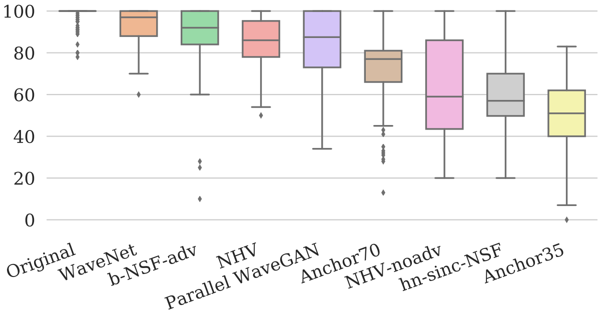

Figure 7: *Box plot of MUSHRA scores*

Table 1: *Mean MUSHRA score with 95% CI in copy synthesis*

Model	MUSHRA Score
Orignial	98.4 ± 0.7
WaveNet	93.0 ± 1.4
b-NSF-adv	91.4 ± 1.6
NHV	85.9 ± 1.9
Parallel WaveGAN	85.0 ± 2.2
Anchor70	71.6 ± 2.5
NHV-noadv	62.7 ± 3.9
hn-sinc-NSF	58.7 ± 2.9
Anchor35	50.0 ± 2.7

Wilcoxon signed-rank test demonstrated that except for two pairs (Parallel WaveGAN and NHV with $p = 0.4$, hn-sinc-NSF and NHV-noadv with $p = 0.3$), all other differences are statistically significant ($p < 0.05$). There is a large performance gap between NHV-noadv and NHV model, showing that adversarial loss functions are essential to obtaining high-quality reconstruction.

3.3.2. *Performance in text-to-speech*

To evaluate the performance of vocoders in text-to-speech, we performed a mean opinion score test. 40 Chinese listeners participated in the test. 21 utterances were randomly selected from the test set and were divided into three parts. Each listener finished one part of the test randomly.

Table 2: *Mean MOS score with 95% CI in text-to-speech*

Model	MOS Score
Original	4.71 ± 0.07
Tacotron2 + hn-sinc-NSF	2.83 ± 0.11
Tacotron2 + b-NSF-adv	3.76 ± 0.10
Tacotron2 + Parallel WaveGAN	3.76 ± 0.12
Tacotron2 + NHV	3.83 ± 0.09

Mann–Whitney U test showed no statistically significant difference between b-NSF-adv, NHV, and Parallel WaveGAN.

3.3.3. *Computational complexity*

We report the required FLOPs per generated sample by different neural vocoders. We do not consider the complexity of activation functions, and computations in feature upsampling and source signal generation. Filters in NHV are assumed to be implemented with FFT. And N point FFT is assumed to cost $5N \log_2 N$ FLOPs.

The Gaussian WaveNet is assumed to have 128 skip channels, 64 residual channels, 24 dilated convolution layers with kernel size set to 3. For b-NSF, Parallel WaveGAN, LPCNet, and MelGAN, hyper-parameters reported in the papers were used for calculation. Further details are provided in the online supplement[4].

Table 3: *FLOPs per sampling point*

Model	FLOPs/sample
b-NSF	$4. \times 10^6$
Parallel WaveGAN	$2. \times 10^6$
Gaussian WaveNet	$2. \times 10^6$
MelGAN	$4. \times 10^5$
LPCNet	1.4×10^5
NHV	1.5×10^4

As NHV only runs at the frame level, its computational complexity is much lower than models involving a neural network running directly on sampling points.

4. Conclusions

This paper proposed the neural homomorphic vocoder, a neural vocoder framework based on the source-filter model. We demonstrated that it is possible to build a highly efficient neural vocoder under the proposed framework capable of generating high-fidelity speech.

For future works, we need to identify causes of speech quality degradation in NHV. We found the performance of NHV sensitive to the structure of the discriminator and the design of reconstruction loss. More experiments with different neural network architectures and reconstruction losses may lead to better performance. Future research also includes evaluating and improving the performance of NHV on different corpora.

5. Acknowledgement

This study was supported by Shanghai Jiao Tong University Scientific and Technological Innovation Funds (YG2020YQ01). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

[4] https://zjlww.github.io/is2020/computational_complexity.html

6. References

[1] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[3] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[4] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, 2018, pp. 2410–2419.

[5] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[6] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[7] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[8] S. Kim, S. Lee, J. Song, J. Kim, and S. Yoon, "FloWaveNet: A generative flow for raw audio," *arXiv preprint arXiv:1811.02155*, 2018.

[9] M. Binkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," *arXiv preprint arXiv:1909.11646*, 2019.

[10] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[11] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. NIPS*, 2019, pp. 14 881–14 892.

[12] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, 2019, pp. 5891–5895.

[13] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram," in *Proc. Interspeech*, 2019, pp. 694–698.

[14] ——, "Waveform generation for text-to-speech synthesis using pitch-synchronous multi-scale generative adversarial networks," in *Proc. ICASSP*, 2019, pp. 6915–6919.

[15] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. ICASSP*, 2019, pp. 5916–5920.

[16] X. Wang and J. Yamagishi, "Neural harmonic-plus-noise waveform model with trainable maximum voice frequency for text-to-speech synthesis," *arXiv preprint arXiv:1908.10256*, 2019.

[17] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. ICLR*, 2020.

[18] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*. Pearson Upper Saddle River, NJ, 2011, vol. 64.

[19] D. W. Griffin and J. S. Lim, "A new model-based speech analysis/synthesis system," in *Proc. ICASSP*, vol. 10, 1985, pp. 513–516.

[20] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[21] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[22] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," in *Proc. ICMC*, 1996, pp. 332–335.

[23] V. Valimaki and A. Huovilainen, "Antialiasing oscillators in subtractive synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 116–125, 2007.

[24] A. V. Oppenheim, "Speech analysis-synthesis system based on homomorphic filtering," *The Journal of the Acoustical Society of America*, vol. 45, no. 2, pp. 458–465, 1969.

[25] I. Saratxaga, I. Hernaez, M. Pucher, E. Navas, and I. Sainz, "Perceptual importance of the phase related information in speech," in *Proc. Interspeech*, 2012, pp. 1448–1451.

[26] P. Mowlaee, J. Kulmer, J. Stahl, and F. Mayer, *Single Channel Phase-Aware Signal Processing in Speech Communication: Theory and Practice*. Wiley Online Library, 2017.

[27] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*. Pearson Education, 2008.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[29] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, 2017, pp. 2813–2821.

[30] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.